

# Exercise 14

Binay Prasanna Jena

October 24th 2020

## Fit a logistic regression model to the binary-classifier-data.csv dataset from the previous assignment.

Problem Statement : Fit a logistic regression model to the binary-classifier-data.csv dataset from the previous assignment.

- What is the accuracy of the logistic regression classifier?
- How does the accuracy of the logistic regression classifier compare to the nearest neighbors algorithm?
- Why is the accuracy of the logistic regression classifier different from that of the nearest neighbors?

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/binay/Documents/GitHub/dsc520/")

## Load the 'caTools' library
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.0.3
```

```
#library(MASS)

## Load the 'data/binary-classifier-data.csv' to
binary_classifier_df <- read.csv("data/binary-classifier-data.csv")
head(binary_classifier_df)
```

```
##   label      x      y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```
summary(binary_classifier_df)
```

```
##      label      x      y
## Min.   :0.000   Min.   : -5.20   Min.    : -4.019
```

```
## 1st Qu.:0.000 1st Qu.: 19.77 1st Qu.: 21.207
## Median :0.000 Median : 41.76 Median : 44.632
## Mean :0.488 Mean : 45.07 Mean : 45.011
## 3rd Qu.:1.000 3rd Qu.: 66.39 3rd Qu.: 68.698
## Max. :1.000 Max. :104.58 Max. :106.896
```

Fit a logistic regression model to the binary-classifier-data.csv dataset from the previous assignment.

```
# Since label is number converting to factors, so that it becomes categorical
binary_classifier_df$label <- as.factor(binary_classifier_df$label)
```

```
# Splitting the dataset for the model into train and test datasets.
myData <- sample.split(binary_classifier_df$label, SplitRatio=0.8)
```

```
train <- subset(binary_classifier_df, myData==TRUE)
test <- subset(binary_classifier_df, myData==FALSE)
```

```
# This model includes all other parameters as dependent
# Using train dataset to generate the model
lrmodel.1 <- glm(label ~ ., family = 'binomial', data = train)
summary(lrmodel.1)
```

```
##
## Call:
## glm(formula = label ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3966  -1.1682  -0.9349   1.1606   1.4210
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.475796   0.131192   3.627 0.000287 ***
## x            -0.002930   0.002057  -1.425 0.154280
## y            -0.008789   0.002103  -4.179 2.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1661.5  on 1198  degrees of freedom
## Residual deviance: 1638.2  on 1196  degrees of freedom
## AIC: 1644.2
##
## Number of Fisher Scoring iterations: 4
```

a. What is the accuracy of the logistic regression classifier?

```
# Using test dataset to see if the model is good
result <- predict(lrmodel.1,test,type = "response")

# result
# validating - putting the actual value and counts of Predicted values in a matrix
# Setting to T if result > 0.5
confmatrix <- table(ActualValue=test$label, PredictedValue = result > 0.5)
confmatrix
```

```
##           PredictedValue
## ActualValue FALSE TRUE
##           0      80   73
##           1      65   81
```

```
# accuracy - Cases where we predicted correctly by Total Predictions
# from matrix, we see when Actual Value is T, confmatrix needs to pick 1,2
# and when F it should pick 2,1
(confmatrix[1,1]+confmatrix[2,2])/sum(confmatrix)
```

```
## [1] 0.5384615
```

So this model shows an accuracy ~55%.

b. How does the accuracy of the logistic regression classifier compare to the nearest neighbors algorithm?

```
library(class)

# Generating knn model with k=1
knnmodel.1 <- knn(train[2:3],test[2:3],k=1,cl=train$label)
summary(knnmodel.1)
```

```
##    0    1
## 148 151
```

```
##create confusion matrix
tab <- table(knnmodel.1,test$label)
```

```
##this function divides the correct predictions by total number of predictions that tell us how accurate
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)
```

```
## [1] 96.32107
```

```
# Running for multiple K Values
for(i in 1:20){
  ##print(paste("Model with K=", i))
  knnmodel.i <- knn(train[2:3],test[2:3],k=i,cl=train$label)
  table.i <- table(knnmodel.i,test$label)
  print(paste("Accuracy for Model with K=", i , " is ", accuracy(table.i)))
}
```

```
## [1] "Accuracy for Model with K= 1 is 96.3210702341137"
## [1] "Accuracy for Model with K= 2 is 96.6555183946488"
## [1] "Accuracy for Model with K= 3 is 96.6555183946488"
## [1] "Accuracy for Model with K= 4 is 96.3210702341137"
## [1] "Accuracy for Model with K= 5 is 97.3244147157191"
## [1] "Accuracy for Model with K= 6 is 96.989966555184"
## [1] "Accuracy for Model with K= 7 is 96.6555183946488"
## [1] "Accuracy for Model with K= 8 is 96.989966555184"
## [1] "Accuracy for Model with K= 9 is 97.3244147157191"
## [1] "Accuracy for Model with K= 10 is 96.3210702341137"
## [1] "Accuracy for Model with K= 11 is 96.989966555184"
## [1] "Accuracy for Model with K= 12 is 96.6555183946488"
## [1] "Accuracy for Model with K= 13 is 96.6555183946488"
## [1] "Accuracy for Model with K= 14 is 96.6555183946488"
## [1] "Accuracy for Model with K= 15 is 96.3210702341137"
## [1] "Accuracy for Model with K= 16 is 97.3244147157191"
## [1] "Accuracy for Model with K= 17 is 95.9866220735786"
## [1] "Accuracy for Model with K= 18 is 96.3210702341137"
## [1] "Accuracy for Model with K= 19 is 95.9866220735786"
## [1] "Accuracy for Model with K= 20 is 95.9866220735786"
```

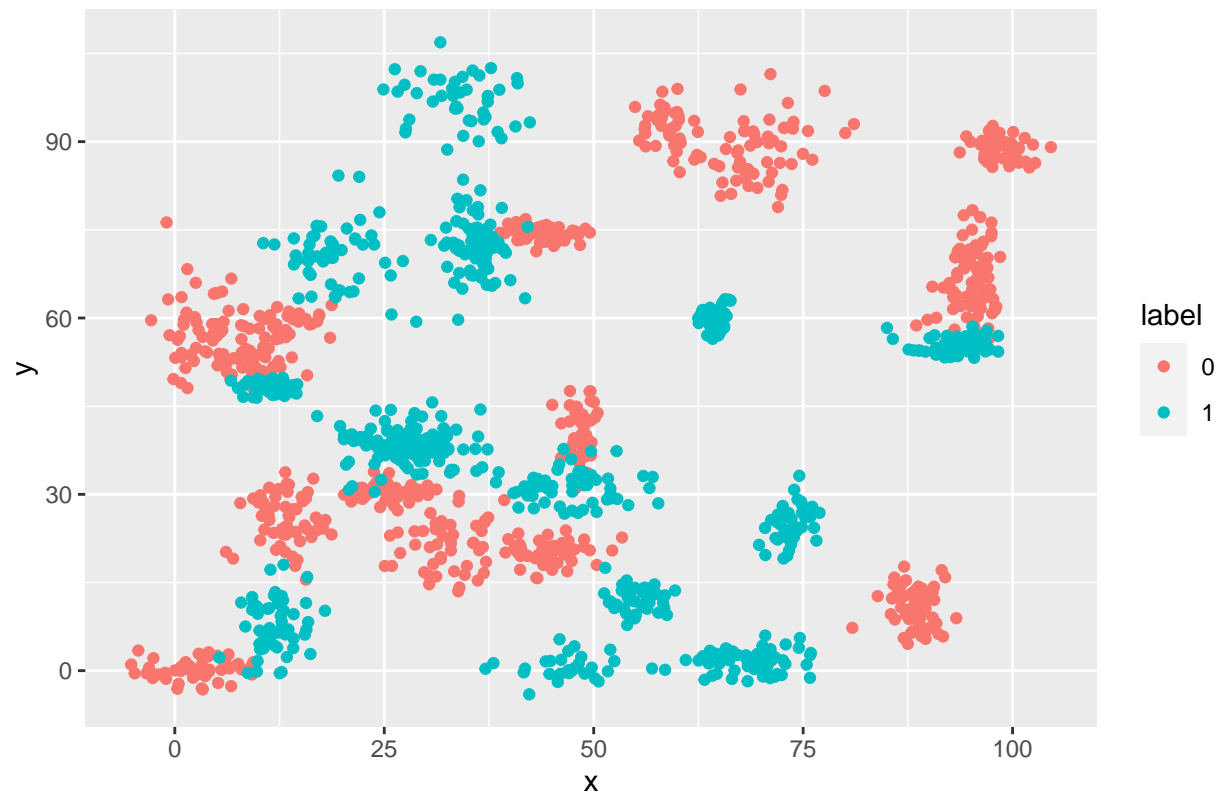
c. Why is the accuracy of the logistic regression classifier different from that of the nearest neighbors?

```
# creating df for plotting the comparison against actuals
df <- binary_classifier_df
df$predict <- predict(lrmodel.1, df,type = "response")

# Adding details to test df for plotting
test$knnpredict <- knn(train[2:3],test[2:3],k=1,cl=train$label)

library(ggplot2)
ggplot(data = binary_classifier_df, aes(y = y, x = x, color = label)) +
  geom_point() + ggtitle("Actual Data")
```

Actual Data

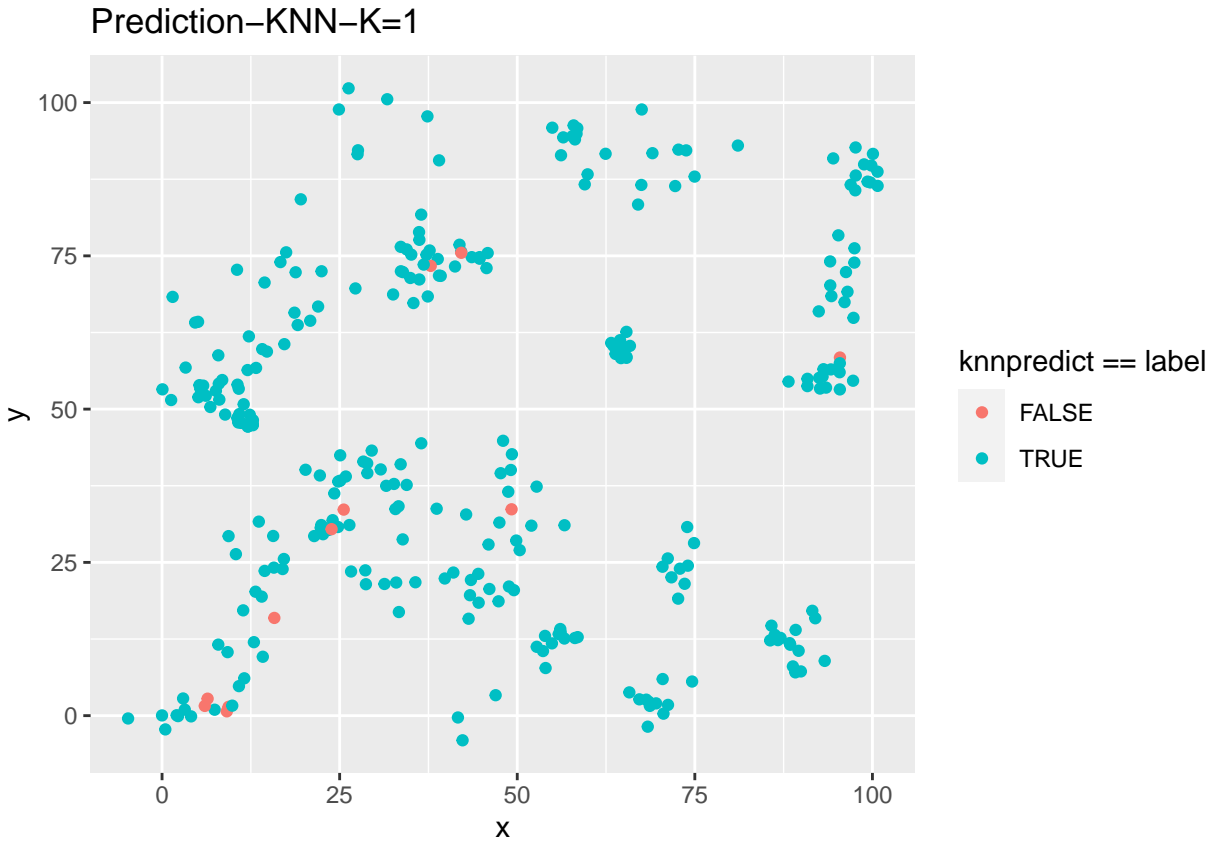


```
ggplot(data = df, aes(y = y, x = x, color = predict>0.5)) +  
  geom_point() + ggtitle("Prediction-Logistic Regression")
```

## Prediction-Logistic Regression



```
ggplot(data = test, aes(y = y, x = x, color = knnpredict == label)) +  
  geom_point() + ggtitle("Prediction-KNN-K=1")
```



If we look at the Actual Data, its hard to divide the data with a line into two separate sections. So the Logistic Regression Model is not a good fit. If we look at “Prediction-KNN-K=1” plot, it seems to make sense as the values are plotted better.

## Reference

<https://towardsdatascience.com/k-nearest-neighbors-algorithm-with-examples-in-r-simply-explained-knn-1f2c88da405c>